

AD-A119 330

DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
FUNCTIONAL DESCRIPTION OF AN INTERACTIVE SIGNAL SYNTHESIS SYSTE--ETC(U)
SEP 82 S BERKOWITZ
DTNSRDC-82/073

UNCLASSIFIED

NL

For I
AD-A
19330

END
DATE
FILMED
10-82
DTIC

AD A119330

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC-82/073	2. GOVT ACCESSION NO. AD-A119330	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) FUNCTIONAL DESCRIPTION OF AN INTERACTIVE SIGNAL SYNTHESIS SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Final
7. AUTHOR(s) S. Berkowitz		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research and Development Center Bethesda, Maryland 20084		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Code 03R22 Washington, D.C. 20360		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Element 61153N; Task 15321; Task Area SR0140301; Work Unit 1808-010
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1982
		13. NUMBER OF PAGES 22
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Digital Signal Synthesis Computers		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Interactive Signal Synthesis System is a user-oriented Fortran pro- gram designed to give the signal analyst the ability to generate controlled time series of arbitrary length and complexity. With the system, one can not only parameterize basic signal functions such as tone, gate, noise, polynomial, and arbitrary data, but also parameterize the shifted sum, damping, modul- ation, and filtering of each function.		

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

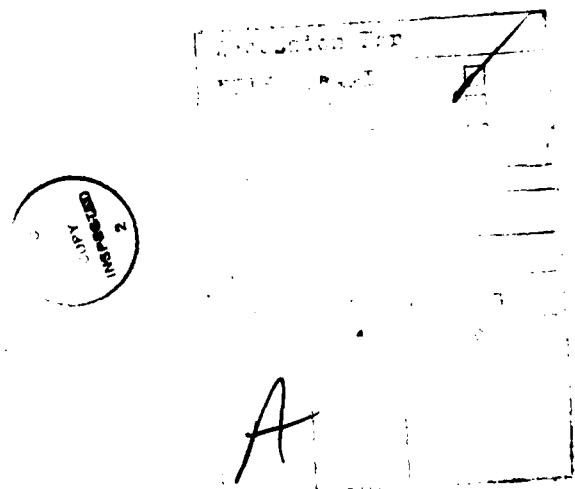
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

	Page
FIGURE AND TABLE	iii
ABSTRACT	1
ADMINISTRATIVE INFORMATION	1
INTRODUCTION	1
FUNCTIONAL DESCRIPTION	2
INFORMATION FLOW	2
PARAMETERS OF ELEMENTARY FUNCTIONS	6
Tone Parameters	6
Gate Parameters	7
Polynomial Parameters	9
Noise Parameters	11
Data Parameters	12
FILTER DESIGN	13
REFERENCES	17

FIGURE AND TABLE

Figure - Information Flow and Control Structure of Parameter Definition and Signal Synthesis	5
Table - Generalized Hamming Window Parameter Variation	14



ABSTRACT

The Interactive Signal Synthesis System is a user-oriented Fortran program designed to give the signal analyst the ability to generate controlled time series of arbitrary length and complexity. With the system, one can not only parameterize basic signal functions such as tone, gate, noise, polynomial, and arbitrary data, but also parameterize the shifted sum, damping, modulation, and filtering of each function.

ADMINISTRATIVE INFORMATION

The work reported here was performed in the Computer Science and Information Systems Division of the Computation, Mathematics, and Logistics Department, within Program Element 61153N, Task Area SRO140301, Task 15321, and Work Unit 1808-010, the Mathematical Sciences Research Program under the sponsorship of NAVSEA 03R22.

INTRODUCTION

The Interactive Signal Synthesis System is a user-oriented Fortran program designed to give the signal analyst the ability to generate controlled time series of arbitrary length and complexity. With the system, one can not only parameterize basic signal functions such as tone, gate, noise, polynomial, and arbitrary data, but also parameterize the shifted sum, damping, modulation, and filtering of each function.

For the convenience of the non-computer-oriented analyst, emphasis has been placed on structuring the interactive user interface with prompts, definitions, and optional echoes. However, for exact formula definitions and discussions of side effects, the user must refer to the material which follows presently. Output signal data may be formatted and scaled for viewing and manipulation by ISPARS (Interactive Signal Pattern Analysis and Recognition System).^{1*} Although the synthesis system has been coded specifically in Fortran for the PDP-11/45, it contains no graphics or assembly language code; its portability therefore is strictly a function of its machine-dependent input/output.

*A complete listing of references is given on page 17.

It is expected that the signal synthesis system will be of use in generating test data for validation of signal analysis programs, in establishing an analysis-by-synthesis procedure for pattern classification applications, and in determining standards for transient waveform definition.

The following functional description of the signal synthesis system outlines the algorithmic aspects of the system. A separate report to be issued as an interactive users' guide will describe the decisions required of a user in executing the program.

FUNCTIONAL DESCRIPTION

INFORMATION FLOW

In the major loop of the Interactive Signal Synthesis System, a sum of sampled, composite functions is computed as follows:

$$g(t) = \sum_i f_i(t) \quad (1)$$

where the time t is related to the sampling period T_s by

$$t = mT_s \quad (m=0,1,\dots) \quad (2)$$

The resultant time series $g(m)$ is placed on disk in INTFILE in integer form and/or on SUMFILE in floating-point form. For viewing the time series on ISPARS, an integer input file is required. The INTFILE serves this purpose and is automatically scaled such that the maximum absolute value in the file equals 8192.

The sum $g(t)$ in Equation (1) may, in fact, be represented in its optionally filtered form $g'(t)$ as follows:

$$g'(t) = \sum_{k=1}^n w_{n-k+1} g(t') \quad (3)$$

where

$$t' = t + [(n+1)/2]T_s - (k-1)T_s$$

The brackets $[,]$ denote "the greatest integer in", and the w_k ($k=1, \dots, n$) are the weights of a symmetric FIR (finite impulse response) filter to be described more carefully later. Equation (3) is then the convolution of a filter $\{w_k\}$ with a function $g(t)$ transposed in time so that the center $k = [(n+1)/2]$ of the symmetric filter (which corresponds to the maximum weight) coincides at time t with $g(t)$. The function $f_i(t)$ of Equation (1) is related to the simpler function $e_i(t)$ by the shift-sum

$$f_i(t) = \sum_{j=1}^R e_i(t - (j-1)T_r) \quad (4)$$

where

R (>0) is the number of recurrences

T_r (>0) is the recurrence shift

$f_i(t) = 0$ for $t < 0$

By judicious use of the damping factor to be associated with every elementary function to be described below, one can generate a repeating sequence of disjoint or overlapping transient signals with the shift-sum mechanism.

The shift-sum $f_i(t)$ of Equation (4) may, in fact, be represented in its optionally filtered form $f'_i(t)$ as follows:

$$f'_i(t) = \sum_{k=1}^P f_i(t'') v_{p-k+1} \quad (5)$$

where

$$t'' = t + ([(p+1)/2] - (k-1)) T_s$$

Equation (5) represents the filtering of a function $f_i(t)$ by a symmetric FIR filter with weights v_k ($k=1, \dots, p$) as in Equation (3). The function $e_i(t)$ of Equation (4) may then be formed from an optional modulation M of the parameters of the elementary functions $d_i(t)$. Formally, one may write

$$e_i(t) = e_i(M, d_i(t)) \quad (6)$$

There are currently five basic types of elementary functions $d_i(t)$:

- tone
- gate
- polynomial
- noise
- data

each of which will be defined in the sections to follow. The modular nature of the system makes it easy to expand the list. In Equation (6), the modulation operator M applied to $d_i(t)$ is implemented by the use of some $d'_i(t)$ selected as any one of the five listed elementary functions from which $d_i(t)$ is itself chosen. Finally, each $d_i(t)$, $d'_i(t)$ is subject to an exponential damping which is included in the elementary function parameter definition.

In terms of the functions just briefly described, the flow of information and interplay of control between parameter definition and time series synthesis is illustrated in the Figure on the next page. In the Figure, labelled lines signify information and unlabelled lines denote control. Thus, the execution sequence is: define the elementary function d (and its damping), its modulation type M (if any), and the modulation elementary function d' (and its damping); at each sample m , evaluate the modulation function d' and apply it to the elementary function d by the operation $M(d)$; shift-sum and optionally define and apply a filter \underline{v} . The parameters and synthesized signal are saved on their respective files. Each iteration will sum the resultant time series onto the INTFILE and/or the SUMFILE with past syntheses of the same program execution and permit the definition and application of a filter \underline{w} . By saving the parameter file, one retains the ability, not only to interactively and selectively modify a parameter set previously defined, but also to resynthesize a signal without interaction. The latter capability is a time-for-space trade-off that permits access to many precisely defined time series without explicitly storing them.

PARAMETERS OF ELEMENTARY FUNCTIONS

In this section, the elementary functions listed previously are defined, with their parameters and modulation characteristics. Each description is self-contained and therefore some redundancy appears.

Tone Parameters

The exponentially damped tone is a fundamental function from which any single-valued, periodic time series of engineering interest may be formed. Nevertheless, certain periodic functions, such as sawtooth or pulse train, which require a truncated approximation to an infinite set of harmonics, are more conveniently and precisely synthesized by more direct definitions, such as the gate and polynomial functions to be described later. Moreover, the user should be aware that a multi-valued function such as a true gate function can be approximated on the current system only by a function whose values are separated by at least one sample. The tone function is defined as follows:

$$d_1(t) = A \exp(-\text{ALPHA } t) \sin(2\pi(\text{FREQ } t + \text{PHASE})) + \text{MEAN} \quad (7)$$

where

A is the amplitude

ALPHA (>0) is the damping constant in Hertz; (ALPHA/2 π
is the resonant bandwidth)

FREQ is the frequency in Hertz; (1/FREQ is the period)

PHASE is the phase as a fraction of 2 π ; i.e., if P is
the phase in radians, then PHASE = P/2 π

MEAN is the mean

One and only one of the parameters amplitude, frequency, or phase is subject to modulation, which has the form

$$p_1(t) = p_1' d_1'(t) \quad (8)$$

where

$p_1(t)$ is one of: $A(t)$, $FREQ(t)/t$, $PHASE(t)$

p_1' is the value of p_1 chosen when

$d_1(t)$ was defined

$d_1'(t)$ is one of the five elementary functions

The option of simultaneous modulation modes is not implemented but is under consideration. The user should note that the frequency modulation defined by Equation (8) is spectral frequency modulation, not the more familiar instantaneous frequency modulation. In order to produce the latter type of modulation, one needs to implement the following relation:

$$FREQ(t) = FREQ \left(\int_0^t d_1'(t) dt + \text{CONSTANT} \right) \quad (9)$$

so that the phase derivative - the instantaneous frequency - is the desired function $d_1'(t)$. One can implement Equation (9) by synthesizing $d_1'(t)$, passing it through a low-pass filter as described later in the section on filter parameters, and frequency modulating $d_1(t)$ by the resultant floating-point SUMFILE (plus a CONSTANT) regarded as a function $d_1'(t)$, as explained later in the section on the data function parameters. If $d_1'(t)$ is itself chosen to be a tone $d_1'(t)$, then instantaneous frequency modulation is implemented simply as phase modulation by delaying the modulating tone by $\pi/2$ and multiplying by the reciprocal of $2\pi FRQ'$, the radial frequency of the modulating tone.

Gate Parameters

As a means for introducing logical gating sequences and gate train synthesis, the gate function is now defined.

$$d_2(t) = \begin{cases} \text{OFFSET} + A \exp(-\text{ALPHA} (t - \text{PHASE } T_F)), \\ \quad t - \text{PHASE } T_F \leq T_G \pmod{T_F} \\ \text{OFFSET}, \quad t - \text{PHASE } T_F > T_G \pmod{T_F} \end{cases} \quad (10)$$

where

OFFSET is the constant offset (i.e., the range mean, not the function mean)

A is the offset to peak amplitude

ALPHA (>0) is the damping constant in Hertz

PHASE is non-negative phase (i.e., delay) as a fraction of the period T_F ; i.e., if D is the delay in seconds $\pmod{T_F}$, then $\text{PHASE} = D/T_F$

T_F is the period; gate frequency is $\text{FRQ} = 1/T_F$

T_G is the gate in seconds (i.e., the on-period duration, not greater than T_F); duty cycle is $T_G/(T_G + T_F)$

One and only one of the parameters amplitude, phase, period, or gate is subject to modulation, which takes the form

$$p_2(t) = p_2' d_1^!(t) \quad (11)$$

where

$p_2(t)$ is one of: $A(t)$, $\text{PHASE}(t)$, $T_F(t)$, $T_G(t)$

p_2' is the constant value of $p_2(t)$ chosen when $d_2(t)$ was specified

$d_1^!(t)$ is one of the five elementary functions

The option of simultaneous modulation modes is not implemented but is under consideration. Note that if $T_G=0$, a single sample is defined as the on-period, thus producing an impulse train.

Polynomial Parameters

A generalized form of the gate function for shaped pulse trains may be synthesized by a polynomial function of order q as follows:

$$d_3(t) = \begin{cases} \text{OFFSET} + A \exp(-\text{ALPHA}(t - \text{PHASE } T_F)) \sum_{i=0}^q a_i (t - \text{PHASE } T_F)^i, \\ \quad t - \text{PHASE } T_F \leq T_G \quad (\text{mod } T_F) \\ \text{OFFSET}, \quad t - \text{PHASE } T_F > T_G \quad (\text{mod } T_F) \end{cases} \quad (12)$$

where

OFFSET is the constant offset

q is the polynomial order

a_i ($i=1, \dots, q$) are the polynomial coefficients

A is the offset to peak amplitude

ALPHA (>0) is the damping constant in Hertz

PHASE is non-negative phase (i.e. delay) expressed as a fraction of the period T_F ; i.e. if D is the delay in seconds (mod T_F), then $\text{PHASE} = D/T_F$

T_F is the period in seconds; frequency is $\text{FRQ} = 1/T_F$

T_G is the gate in seconds (i.e. the on-period duration, not greater than T_F); duty cycle is $T_G/(T_G + T_F)$

One and only one of the parameters amplitude, phase, period, or gate is subject to modulation, which has the following form:

$$p_3(t) = p_3^i d_1^i(t) \quad (13)$$

where

$p_3(t)$ is one of: $A(t)$, $\text{PHASE}(t)$, $T_F(t)$, $T_G(t)$

p_3^i is the constant value of $p_3(t)$ chosen when $d_3(t)$ was specified;

$d_1^i(t)$ is one of the five elementary functions

When Equation (12) is applied to the generation of a polynomial of specific order, care must be taken to choose the a_i properly. Some examples follow. In each example, the function will start at a level X_{on} at t_0 , and turn off at t_T by switching to the level X_{off} . Thus, from Equation (12), one can immediately write

$$a_0 = (X_{on} - X_{off})/2 \quad (14)$$

$$OFFSET = (X_{on} + X_{off})/2 \quad (15)$$

Example 1. Ramp sequence. - If a first order polynomial is chosen as the basic pulse shape, a gated ramp sequence results. The ramp is defined as rising or falling to a target value X_{tar} in a duration time T_{dur} . The equation for the first order coefficient is then as follows:

$$a_1 = (X_{tar} - X_{on})/T_{dur} \quad (16)$$

Example 2. Triangle sequence. - A gated triangular wave results from the synthesis of the sum of two ramp sequences. If the peak has a value X_{tar} which is reached in duration time T_{dur} from the origin, then the value for a_1 is the same as in Equation (16). If the triangle terminates at a terminal value X_{ter} at time sT_{dur} from the origin, then the second ramp begins at time T_{dur} at level zero and has the slope

$$a'_1 = (X_{ter} - sX_{tar} + (s-1)X_{on})/(s-1)T_{dur} \quad (17)$$

Example 3. Parabolic sequence. - A third order polynomial chosen as the basic pulse shape will produce a gated parabolic sequence. If the parabola rises or falls to successive values X_{med} , X_{tar} at time intervals T_1 , T_2 , respectively, the equations for the two higher order coefficients are as follows: (Note that the total gate duration T_{dur} is the sum of the intervals T_1 , T_2 just mentioned.)

$$a_1 = (T_1^2(X_{on} - X_{tar}) + T_{dur}^2(X_{med} - X_{on})) / T_1 T_2 (T_1 + T_2) \quad (18)$$

$$a_2 = (T_1 X_{tar} + T_2 X_{on} - T_{dur} X_{med}) / T_1 T_2 (T_1 + T_2) \quad (19)$$

If the gated parabola sequence were being used to frequency modulate a tone, the coefficient a_2 would have to be adjusted for the fact that the instantaneous frequency would be computed by differentiating the polynomial, thus doubling the contribution of the a_2 term.

Noise Parameters

A representation of random behavior may be introduced in the synthesized signal by drawing samples from either uniform or Gaussian distributions.

In the uniform case, a random variable RAN uniform on (0,1) is derived from a FORTRAN library function that requires two user-supplied integers to initiate the generation. The uniform noise function with mean MEAN and standard deviation A SIGMA is then defined as follows:

$$d_{4u}(t) = A \text{ SIGMA} \sqrt{12} \exp(-\text{ALPHA } t) (\text{RAN}(t) - .5) + \text{MEAN} \quad (20)$$

where

A is the amplitude gain

ALPHA (>0) is the damping constant in Hertz

RAN(t) is a uniform random variable on (0,1) computed recursively from RAN(t-1)

SIGMA is the standard deviation for A=1

MEAN is the mean

In the Gaussian case, a univariate distribution may be derived from two independent uniform distributions, as explained in a paper by Franklin.² The appropriate time series is divided into even and odd components as follows:

$$d_{4g}(2t) = A \text{ SIGMA } \exp(-\text{ALPHA } 2t) \sqrt{-2 \ln(\text{RANA}(t))} \cos(\text{RANB}(t)) + \text{MEAN} \quad (21)$$

$$d_{4g}(2t+1) = A \text{ SIGMA } \exp(-\text{ALPHA}(2t+1)) \sqrt{-2 \ln(\text{RANA}(t))} \sin(\text{RANB}(t)) + \text{MEAN} \quad (22)$$

where

A, SIGMA, ALPHA, MEAN are as above for uniform noise;
 RANA(t), RANB(t) are independent random variables on (0,1),
 computed recursively from RANA(t-1), RANB(t-1),
 respectively.

Only amplitude modulation is available for the noise functions, in the form

$$A(t) = A' d_1'(t) \quad (23)$$

where

A' is the defined amplitude of $d_4(t)$
 $d_1'(t)$ is one of the five elementary functions

Data Parameters

The final elementary function is in a sense arbitrary. Samples generated either empirically or by previous syntheses may be read from any integer or floating-point user-defined data file. This file must have been created using the PDP 11/45 SYSLIB function IWRITW. The form of the data function is as follows:

$$d_5(t) = A \text{ DATA}(t) + \text{OFFSET} \quad (24)$$

where

A is the amplitude gain
 DATA(t) is the data
 OFFSET is the offset

Only amplitude modulation is available for the data function, in the form

$$A(t) = Ad_1'(t) \quad (25)$$

where

A is the defined amplitude gain of $d_5(t)$

$d_1'(t)$ is one of the five elementary functions

FILTER DESIGN

As indicated in the Figure, the signal being synthesized may be filtered after shift-sum or after the iterate sum. The total filter may be a composite of up to 30 lowpass, bandpass, bandstop, or highpass filters, each of whose weight vectors \underline{w}_i is designed using canned software written by Rabiner, McGonegal, and Paul. This software has been published by the IEEE³ in a more comprehensive filter design package than that offered here. The composite filter vector is formed by a normalized sum of the component filter weights w_{ij} against gains g_i , as follows:

$$\underline{w} = \sum_{i=1}^n g_i \underline{w}_i / \sum_{i=1}^n g_i \quad (26)$$

where the filter length n is an integer lying between 3 and 896. For highpass or bandpass filters, n must be odd. The component filters are symmetric, linear phase FIR digital filters designed on the basis of cut-off parameters and against the following windows:

- rectangular
- triangular
- Hamming

- generalized Hamming
- Hanning
- Chebyshev
- Kaiser-Bessel

The exact forms of the windows may be found in the IEEE document³ mentioned earlier, page 5.2-2. Here, however, the parameters, if any, associated with each window and their effects are only briefly summarized. Notwithstanding the theory, the user will find it necessary to experiment with window selection and parameter values to meet his needs. Note that specification of a desired gain parameter and appropriate frequency cutoffs is common to all the filter designs.

The rectangular, triangular, Hamming, and Hanning windows have no user-defined parameters other than gain and cutoffs.

The generalized Hamming window has one parameter, ALPHA, ranging over $\langle 0,1 \rangle$. The following table illustrates the effect of varying ALPHA:

TABLE 1 - GENERALIZED HAMMING WINDOW PARAMETER VARIATION

ALPHA =	WINDOW =
1.0	rectangular
0.54	Hamming
0.5	Hanning
0.0	$\cos(2\pi i/n)$ for $-n \leq 4i \leq n$

For the Chebyshev window, in addition to specifying the frequency cutoff and desired passband gain, one must specify two of three parameters from which the third is calculated. The parameters are: filter length n , transition width DF , and ripple DP . The transition width is the length of a frequency band between the passband and stopband. These bands are defined as follows. Let G denote the desired passband gain. Then the frequency for which the computed gain first falls below $G-DP$ is called the passband cutoff. Similarly, the frequency for which the gain first falls to DP is called the stopband cutoff. The difference between the passband and stopband cutoffs is called the transition width DF . It is the passband cutoff which

will have been specified as the desired frequency cutoff by the user. The passband ripple is defined to be the maximum deviation in decibels above the desired gain in the passband. Similarly, the stopband ripple is defined to be the maximum deviation from zero gain in the stopband. In the present design, both passband and stopband ripples are defined equal to DP. For given filter length n , DF and DP vary inversely.

The Kaiser-Bessel window has one parameter ATT which is the desired minimum stopband attenuation in decibels.

REFERENCES

1. Berkowitz, S., et al., "ISPARS - Interactive Signal Pattern Analysis and Recognition System," in publication.
2. Franklin, J., "Numerical Simulation of Stationary and Non-stationary Gaussian Random Processes," SIAM Review Vol. 7, No. 1, pp. 65-80, (Jan., 1965).
3. Weinstein, C.J., et al., eds., "Programs for Digital Signal Processing," IEEE Press, New York, 1979.

INITIAL DISTRIBUTION

Copies

2 NAVSEA
 1 03R22
 1 55X4

12 DTIC

CENTER DISTRIBUTION

Copies	Code	Name
1	1576	W. Smith
1	1606	S. de los Santos
1	173	A. Stavovy
1	18	G. Gleissner
2	1809.3	D. Harris
1	182	A. Camara
20	1824	S. Berkowitz
1	184	J. Schot
1	193	J. O'Donnell
1	1925	G. Clark
1	273	W. Blumberg
10	5211.1	Reports Distribution
1	522.1	Unclassified Lib (C)
1	522.2	Unclassified Lib (A)
1	93	

**LATE
LME**